

SAKARYA ÜNİVERSİTESİ



SAKARYA ÜNİVERSİTESİ

Öğrencinin Adı ve Soyadı	Selim Can ERMIŞ
Doğum Yeri ve Yılı	Ümraniye - 1995
Fakülte Numarası	
Bölümü	Bilgisayar Mühendisliği
Yaptığı Staj Dalı	Yazılım



STAJ BİLGİLERİ

İşyeri Adı	İspark İstanbul
Adresi	Yavneuler Mahallesi Alemdağ Caddesi Kocatepe Sokak No:4
Telefon Numarası	0216 635 00 00
Staj Başlama Tarihi	06.08.2018
Staj Bitiş Tarihi	10.09.2018
Staj Süresi (gün)	20

İşveren veya Yetkilisinin

Adı ve Soyadı : Hakan YILMAZ
 İmzası ve Kaşe
 Bilgi İşlem ve Otomasyon
 Şefi

Önemli Açıklamalar

Öğrencilerin staj yapması zorunludur.
 Staj süresince işyerine devam zorunludur.

Öğrencinin aşağıda belirtilen kurumda staj yapması uygun görülmüştür.

Staj Yeri:

Staj Komisyonu Üyesinin

Adı ve Soyadı :

İmzası :

İŞYERİ İÇİN ÖNEMLİ NOT

İşyerlerinin öğrenci ile ilgili sorunlarının olması durumunda aşağıdaki e-posta adresine şikayetlerini bildirebilirler.

e-posta @sakarya.edu.tr

Öncelikle veritabanı ve SQL server konularının tanıtılması için araştırmalar yapıldı. Ardından Microsoft SQL Server bilgisayara kuruldu.

Veritabanı nedir?

İstenildiği zaman ulaşılacak türde bilgi ve verilerin düzenli bir şekilde saklandığı ortadır. MySQL, MsSQL ve ORACLE gibi çeşitli veritabanları kullanılmaktadır. Ben Microsoft SQL Server kullandım.

MSSQL seçiminin nedenleri araştırıldı. Hem web hem de masaüstü uygulamalarında rahatlıkla kullanılabilmesi bir avantajdır. MSSQL bir sunucu üzerinde çalışmaktadır. Oluşturduğumuz veritabanı arasında bir sunucu üzerinde oluşturulur. Bu sayede rahatlıkla taşınabilir.

Access de bir veritabanı yönetim sistemi dir. Ancak sunucu üzerinde çalışmayı istemci ya da kullanıcı tarafından olşturulabilir.

MSSQL de herkes bir veritabanına bağlanır. Bu veritabanı ortaktır. Makineden kontrol esastır. Access bu konuda eksik kalırktadır.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Otomasyon
Şefi

SQL komutları ile ilgili bilgiler alındı ve bu komutlar ile neler yapabileceğim öğrenildi.

Öncelikle veritabanı oluşturma ve oluşturulan veritabanına bağlanma ile ilgili komutlar öğrenildi.

Veritabanı oluşturma: Create komutu kullanılır.

- Create DATABASE OtoparkOtomasyonu

Veritabanına bağlanma: Use komutu kullanılır.

- Use OtparkOtomasyonu

Sonrasında veri tabanında kullanılabilecek veri tiplerine bakıldı.

Metinsel veri tipleri: char, varchar, text, nchar, nvarchar.

- varchar(20) dediğimizde 20 karakterlik yer açılır ve içerisinde 5 karakter yazılısa bile hala 20 karakterlik yer tutulur.

- nvarchar(20) dediğimizde ise yine 20 karakter max olabilmeli ancak 5 karakter yazılırsa sadece 5 karakterlik yer tutulur.

Numerik veri tipleri: bit, tinyint, smallint, int, bigint, float, real.

- Čarşık olmak aracın otomatik olup olmaması ile ilgili bir durum sıfatı ekledik. Durum 0 ise arac otomatik değil, 1 ise otomatik olsun. Bu durum sıfatının veri tipini bit yapmanızla daha rahatlıkla yeticektir.

Diger veri tipleri = Money, date, datetime, time, binary vs..

- date ile sadece tarihleri tutabiliniz

- datetime ile ise hem tarih hem de saatleri tutabiliniz.

- Money, parayı bir veritipidir. Ve oldukça kolaylıdır.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin
Hakan YILMAZ
Bilgi İşlem Ve Otomasyon
Şefi

Tablo oluşturma işlemleri yapıldı ve bazı özel terminler ögrenildi.

```

CREATE DATABASE sirket;
USE sirket;
CREATE TABLE personel(
    personel-no INT IDENTITY(1,1) PRIMARY KEY,
    ad VARCHAR(25) NOT NULL,
    soyad VARCHAR(25) NOT NULL,
    cinsiyet CHAR(1) NOT NULL,
    dogum-tarihi DATE NOT NULL,
    dogum-yeri VARCHAR(25) NOT NULL,
    calisma-saati TINYINT NOT NULL,
    Maas MONEY NOT NULL, );
  
```

* IDENTITY(1,1) = Bu sadece personel no 1'den başlar ve otomatik her kayıtta sonra 1'er 1'er artar.

* PRIMARY KEY = Birincil anahtar olarak tanımlanır ve her tabloda bir tane olmak zorundadır.

* NOT NULL = Yazılan sectionun içi boş olamaz eğer boş bırakılırsa kayıt yapılmaz.

Ardından oluşturulan tabloya eklenen yepme komutu da INSERT INTO kullanıldı.

```
INSERT INTO personel VALUES ('Selim Cen', 'Ermis', 'e',
'1995.11.16', 'umurigie', '10', '2500');
```

Oluşturulan tablo silmek için de DROP komutu kullanılır.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ

Bilgi İşlem ve Dijitalleşme
Şefi

SQL komutları ile sorgulama işlemleri yapıldı. Sorgulamada yardım teknikler kullanarak pratik yapıldı.

Select: Select * from table-adi;

Distinct: Yinelenen kayıtlar alınmak için.

Select DISTINCT sutun1, from table-adi;

Where : Sat lagna işlemi

Select DISTINCT sutun1, From table-adi

where Sart-ifadesi;

AND / OR, BETWEEN, LIKE, COUNT, SUM, AVG.

Select sutun1, sutun2 from table-adi

where sutun-adi BETWEEN deger1 AND deger2;

ORDER BY : Sıralama yapmak için. (ASC/DESC)

GROUP BY : Gruplandırma yapmak için.

Select COUNT(sutun-adi) From table-adi

where Versa-Sort;

Truncate Table : Tablonun içindeki tüm verileri siler.

Drop table : Direk olarak tabloyu siler.

Select KitapYayinevi COUNT(KitapAd) as 'Kitap Sayısı',

Sum (KitapFiyat) as 'Alacak',

Sum (Vergi) as 'Vererek',

(Sum(KitapFiyat) - Sum(Vergi)) as 'Net Bakiye',

From Kitap group by KitapYayinevi

having count(KitapAd) >= 4

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Otomasyon
Şefi

Yapılan İş: Veritabanı Taraması, Oluşturma, Eklene, Silme, Dözenleme Tarih: 10 / 08 / 2018

Bir şirket veri tabanı sıfırda tablolar ile oluşturuldu. Tablolar oluşturulduktan sonra eklene, silme, güncelleme işlemleri ile soru işaretleri ile pratik yapıldı. Tablolar arası referans alma işlemleri gerçekleştirildi.

Oluşturulan şirket veri tabanında; birim, unvan, il, ilce, proje, personel, cüzük, görevlendirme tabloları oluşturuldu.

il adları UNIQUE olarak kaydedildi. İlçeler ise ilerden referans alındı. oncek ilçeler UNIQUE yazılmadı çünkü birack ilde aynı isimde ilçe bulunabilir.

Personel tablosundaki birim ismi birim tablosunda, unvan ismi unvan tablosundan, doğum-yeri ise ilçe tablosundan referans olarak yapıldı.

Sıfırda bir veri tabanı oluşturmak tamamen halledilmiș oldu. Oluşturulan bu veri tabanının ilişkisel diyagramı yapıldı. Oluşturulan bu tablo üzerinden Delete, Update, Create, INSERT INTO ile örnekler yapıldı.

update : update table_adi;

SET sutun1=deger1, sutun2=deger2

[WHERE Sart];

Delete : Delete from table_adi;

[WHERE Sart];

INSERT INTO : INSERT INTO table_adi(sutun1,sutun2...sutunN)

VALUES (deger1, deger2 degerN)

Öğrencinin İmzası

Staj Amirinin

Hakan YILMAZ

Bilgi İşlem ve Otomasyon,

Şefi

Adı Soyadı :
İmzası :

Yapılan İş: İş kuralları oluşturma

Tarih: 13 / 08 / 2018

İkinci haftanın başında SQL'den veri çekme, ekleme, güncelleme, silme işlevlerine sahip veren bir Otopark Otomasyonu hazırlamasına karar verildi. C# dilinde bir Windows Form uygulaması olmasına rağmen windows visual studio kullanıldı.

Öncelikle ilk gün taraman iş kuralları ve yazılım yapım dengesine uygun bir planlara yaradıldı. En ince ayrıntısına kadar planlanarak yarıldı.

İŞ KURALLARI

Kullanıcı Girişisi:

- * Uygulamada kullanıcı giriş paneli gelicektir.
- * Kullanıcı adı ve parola ile giriş yapılacaktır.
- * Yatırış giriş sonucunda hata mesajı verilecektir.
- * Yatırış giriş sonucunda hata mesajı verilecektir.
- * Parolanın göster sekmesi bulunacaktır.
- * Kullanıcı girişinin sol alt köşesinde hakkımızda, iletişim ve oturum kapat butonları bulunacaktır.
- * Boşaltı bir şekilde giriş yapılrsa ana sayfa ekranı açılacaktır.

Anasayfa:

- * Anasayfa ekranının altında oturum kapat sekmesi olacaktır.
- * Oturum kapatıldığında kullanıcı giriş sayfasına geri dönercektir.
- * Müşteriler sekmesi bulunmaktadır.
- * Otopark durum sekmesi bulunmaktadır.
- * Kasa sekmesi bulunmaktadır.
- * Bu seküler PictureBox'lar ile oluşturulacaktır.
- * Cursor eklenecektir.
- * PictureBox'la event eklenerek tıklanınca yeni sayfa açılacaktır.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

MÜŞTERİ HİZMETLERİ
Bilgi İşlem ve Otomasyon
Şefi

Yapılan İş: İŞ KURULLARI DÜZLEŞTİRME

Tarih: 13/10/2018

Müşteriler:

- * Müşteriler sekmesinin içerisinde Müşteri Listesi, Müşteri ekleme ve Müşteri düzenlene sekmeseleri bulunacaktır.
- * Bu sekülerde yine pictureBox'larla oluşturulacaktır.
- * Genel butonu olacaktır.
- * Müşteri listesine tıkladığında yeni bir formda veritabanından alınan veriler datagrid ile ekranaya verilecektir.
- * İstenilen bilgiler: ad, soyad, TC, mail, kayıt tarihi, adres, plaka, moka, model ve renk olacaktır.
- * Müşteri düzenlene sekmesine girdiğinde Müşteri Listesi açılacaktır. Burdan herhangibir hücreye tıklanması sonucu düzenlene formu ekranaya gelecektir.
- * Hücreye tıkladığında bilgiler gerekli textbox'larla gelecektir.
- * Buradan isteğe bağlı olarak güncelleme veya silme yapılabilir.

Kosar:

- * Kosarınla para veritabanında eklederek toplam dairesekilde ekranaya verilecektir.
- * Seçilen yıl ve cıza göre en çok park hizmetini kullanan araç schibinin konu bilgileri ekranaya verilecektir. Kesinlikle TC numarası ile paylaşılacaktır.
- * Araç çıkışlarında ödeme alınacaktır. Buın için yeri Roma gonderilecektir. Ödeme ismini suan için nakit ve kredi olacaktır. Ancak her zaman geliştirilmeye uygun bir yazılım olacaktır.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Oluşturma
Şefi

Yapılan İş: iş karotları değişturma

Tarih: 13 / 108 / 2018

Otopark Durum:

- * Otopark Durum formu açıldığında buttonlardan oluşan bir park düzeli karışımıza gelicektir.
- * 4 blok ve her blokta 9 cep bulunacaktır. Bu ceplerin isimleri A1, D5, C3, B3 gibi isimlendirilecektir.
- * Cepler boş olduğunda buton rengi "gold" dolu olduğunda ise "darkorange" olacaktır.
- * Ceplerin dolu/bağı sergisi veritabanı üzerinden yapılmaktır.
- * Formun sağ bölümünde kaa cebin dolu/bağı olduğu belirtilecektir.
- * istenildiğinde orası faga dönmek için geri butonu olacaktır.
- * Her cep buton olacığı için orası girişinde seçilen butona tıklanacaktır. Bu sayede orası giriş formuna yönlendirilecektir.
- * Anadak bu sadece 'gold' renkli butonlar için geçerlidir.
- * Eğer buton 'gold' ise adanın orası giriş sayfasında orası polşeri ve giriş zamanı (DateTime.Now) ile otomatik olacaktır.
- * Öncelikle orası plaka kontrol ile müşterinin kayıt durumu kontrol edilecektir.
- * Kayıtlı bulunmayan kullaniciların girişine izin verilmeyecektir ve kayıt almalar için kayıt sayfasına yönlendirilecektirler.
- * Kayıtlı ise kullanicı (müşteri) bilgileri ekran'a gelecek ve giriş yap butonuna tıklandığında bu bilgiler veritabanına ayrı bir tablet olarak kayıtlı edilecektir. Bu sayede o cepte hangi macin olduğu bilinicektir.
- * Buton 'darkorange' ise orası çıkış sayfasına yönlendirilecektir.
- * Müşteri bilgileri ekran'a gelicektir, çıkış yapmadan önce.
- * Aynı zamanda kaa saat depon kullandığı ve sonunda ne kadar ücret ödenecüğü de ekran'a gelecektir.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakem YAZ
Bilgi İşlem ve İletişim
Şefi

Yapılan İş: Analiz ve Veritabanı Tasarımı

Tarih: 14 / 08 / 2018

Analiz aşamasına geçildiğinde ve uygulanıyor gerekli ekranlar
yapılmıştır.

Analiz aşamasının olmasına ve adminin yetkililerine konu verilmiştir.
Reportlar, grafik ve kullanıcı paneli kontrolü. Sadece admin
yapabilecektir. Admin bazı istatistik bilgilerde de erişebilirktir.

Otopark sisteminde cepler olusturma fikrinin nedeni kişiye
özel otopark yer olusturmak ve oracın nerede olduğunu iki tarafında,
kesen dolap bilmesi. Bu sistem ile araç sahibi istediği yere
arkasını birekip gitmeyecek.

Eski ve klasik sisteme otoparkın bir kapasitesi olur ve
kapasite dolana kadar araç girişi alınır. Kapasite dolduğunda
araç durdurulur. Araç otopark içinde dolaşmadığı
kontrolü yapılabılır araç nerede olduğunu kontrolü olmazdır.

Bu yeni sisteme bir siniraga bilet alır gibi yeriini
önceden seçicez araç belirli saatle yaradı ile ola
aldığımız bilette yaran yer dışında bir yere park ederkeniz?

Bu ekranlar yapıldıktan sonra dörtlü ve ayrıntılı olarak veritabanı
oluşturuldu. Veri tabanında kullanıcılar, müşteriler, park durum, kasa
ve Odeme Alma tablaları bulundu. Veri tabanı isimleri büyük
ölçüde bitmiş oldu.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Otonomasyon
Şefi

Yapılan İş: Bağlantı sayfası oluşturma

Tarih: 15 / 08 / 2018

Veritabanı işlemleri büyük ölçüde halledildikten sonra otak visual studio ortamına giriş yapıldı. Yazılım dilimizi C# olarak seçtiğimizde.

Tasarım kısmına geçmeden önce bağlantı işlemini halletmemiz gereklidir. Bunun için uygulama da yeni bir SqlBaglantim adında bir class oluşturduk.

Amaçım her form sayfasında bilgisayar adıyla bir bağlantı oluşturmak yerine gordüğüm bu classı kullanmaktı.

SqlBaglantim.cs:

```
using System.Data.SqlClient;  
namespace OtoparkOtomasyonu  
{  
    public class SqlBaglantim  
    {  
        public SqlConnection baglan()  
        {  
            SqlConnection baglan = new SqlConnection(@"Data  
Source=DESKTOP-60TJEF6;Initial Catalog=Otopark;  
Integrated Security=True");  
            baglan.Open();  
            return baglan;  
        }  
    }  
}
```

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Teknoloji
Şefi

Yapılan İş: Form Özellikleri ve Form Tasarımı (Giriş, Anasayfa) Tarih: 16/08/2018

Form tasarımlı işlerimle başlamıştım. Öncelik olarak kullanacağım formların özelliklerini belirledim. Logoları ve pictureBoxları, internetten buldum. PictureBoxlarda kullanacağım fotosları, icon haline getirdim. Yaklaşık olarak 30 tane icon kullanılmasına karar verdim.

Form Özellikleri:

AutoSizeMode = None;

BackColor = Gold

Cursor = Arrow

FormBorderStyle = None

Size = 700,500

StartPosition = CenterScreen

Bu özellikler kullanacağım tüm formlar için ortak olacaktır.

- Giriş formunu oluşturdum. (frmGiris)

* içerisinde Logo, kullanıcı adı ve parola için iki textBox, giriş için bir pictureBox ve parolayı gester adında bir checkBox bulunacaktır.

checkBox:

```
private void chParola_CheckedChanged(EventArgs e)
{
    if (chParola.Checked)
        txtParola.PasswordChar = '\0';
    else
        txtParola.PasswordChar = '\*';
}
```

- Anasayfa formunu oluşturdum. (frmAnasayfa)

* içerisinde 4 tane pictureBox vardır. Otopark Durum, Kasa İşlem, Abone işlem ve Atama.

Oturumu kapat seçmesi bulunmaktadır.

Öğrencinin İmzası

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Öğrenme
Şefi

Adı Soyadı :
İmzası :

Yapılan İş: Hakkımızda ve İletişim Formları

Tarih: 17/08/2018

Giriş ve Anasayfa formlarının ordindan hakkımızda ve iletişim formlarını da oluşturduum. Formların birbirlerine geçişini sağlayarak kodları yordam.

```
frmAnasayfa frm = new frmAnasayfa();
frm.show();
this.dispose();
```

Bu izlemelelerin ordindan veritabanı ile bağlantılı kullanıcı adı ve parola sorgusunu gerçeklestirdim. Eğer veri tabanından alınan kullanıcı adı ve parola ile eşleşme sağlanmasa ekranı hata mesajı verdiğim. Boşlık bir eşleşme olursa ise yukarıdaki kod ile anasayfaya geçiş sağlanır.

Giriş sayfasına oturumu başlatıp genel denetleme imkanı oluşturduum. Giriş sayfasının sağ alt köşesine hakkımızda ve iletişim için pictureboxlar yerleştirdim.

İletişim sayfasının içerisinde ulaşım bilgileri ve sosyal medya hesaplarına geçiş için link labeller koymu.

Hakkımızda sayfasında ise sadece bilgi verici labeller bulnduruldu.

Aynı zamanda kullanıcı uygulaması, kapatın diye bir buton koymu.

Tooltipler kullanarak fotoğraflar ekrana gelince cursor=arrow yerine cursor=hand olduktan sonra bilgi vericekler küçük tipler koymu.

İkinci haftanın son gününde yaptığı 4 formu ve hazırladığım veritabanını tekrar kontrol ederek görev bitirdim ve bayan töphiline aktardım.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Kütüphane Şefi

Yapılan İş:

Tarih: 17 / 08 / 2018



Öğrencinin İmzası

A handwritten signature in blue ink, appearing to be the name of the student.

Adı Soyadı :
İmzası :

Staj Amirinin
Hakan YILMAZ
Bilgi İşlem ve Otopark
Şefi

Yapılan İş: Müşteri Sayfası, taban (listelere, eklene, düzenlene) Tarih: 27/08/2018

Tatil dengesi arastırın oturumka girisi için müsteri kaydı zorunlu olduğundan bu işe başladım.

Müşteriler, Müşteriler listesi, müsteri eklemeye ve müsteri düzenlemeye için 4 ayrı form oluştururdum.

Müşteriler Listesine veriler veri tabanından datagrid ile gelerek şekilde ayordadım.

Müsteri eklemeye kısmından gerekli bilgileri girdikten sonra veri tabanına kaydettim. Bunun için 'Insert into' kodunu kullandım.

Örnek code:

```
SqlCommand cmd = new SqlCommand("Insert into Musteriler(C  
MusteriAdSoyad, MusteriTC) Values (@p1, @p2)", con.baglanti());  
cmd.Parameters.AddWithValue("@p1", txtMusteriAd.Text);  
cmd.Parameters.AddWithValue("@p2", txtMusteriTC.Text);  
cmd.ExecuteNonQuery();  
con.baglanti().Close();  
MessageBox.Show("Kayıt Başarılı.");
```

Müsteri düzenlemeye sayfasında ise 'delete' ve 'update' komutlarını kullandım. Bu sayfada veri tabanına bağlanarak değişiklik yapıp kaydettim.

Güncelleme =

```
SqlCommand cmd = new SqlCommand("Update Musteriler set  
MusteriAdSoyad=@p2 where Musteriid=@p1", con.baglanti());
```

Silme =

```
SqlCommand cmd = new SqlCommand("Delete from Musteriler  
where Musteriid=@p1", con.baglanti());
```

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Odamasının
Şefi

Yapılan İş:

Tarih: 27/08/2018

Kişisel Bilgiler

Ad Soyad: _____
T.C.: _____
Telefon No: () ____-____
Mail: _____
Kayıt Tarihi: _____._____._____
Adres: _____

Araç Bilgileri

Plaka: _____
Marka: _____
Model: _____
Renk: _____

Geri **Kaydet**

ISPARK
İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.

Kişisel Bilgiler

Ad Soyad: _____
T.C.: _____
Telefon No: () ____-____
Mail: _____
Kayıt Tarihi: _____._____._____
Adres: _____
Id: _____

Araç Bilgileri

Plaka: _____
Marka: _____
Model: _____
Renk: _____

Güncelle **Sil** **Geri**

ISPARK
İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.

Öğrencinin İmzası

Adı Soyadı : _____
İmzası : _____

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Otopark
Şefi

Yapılan İş: Otopark Durum ile blok tesorim

Tarih: 28/10/2018

Otopark durum formu için ekstra bir blok ve her blok için 9 cep blok şekilde 36 adet buton koydu ve renklerini 'gold' olarak ayırdı.
Form load olduğunda yarışım kod sayesinde dolu olan bir cep varsa
o cep 'darkorange' blok renk deşistiyor. Aynı zamanda tıkladığında
acil blok olan yeni formu da deşistişmiş oluyor. Cep rengi 'gold' iken
araa giriş; rengi 'darkorange' iken ise araa çıkış formuna geçiş
seçiliyordur.

Araa giriş çıkış formlarını hazırladın öncek kodlama istemelerini
diğer sine birdettim.

Butoruların hepsine tıkla ile obgack form atadığım için tıklayıcı
aktif hale getirdim.

Araa giriş sayfasına gittiğim zaman o butonun ismini park yerini
blok kabul edecek ve yarışır textBox deşistişmesin diye
enabled = false blok deşistiştirdim.

Otopark Durum formu load olduğunda aynı zamanda butonların
renklerine göre sayılmaya başlanır bu sayıları göre soñ kenara dolu/bos
sayılarını verecektir.

Bu soruya renklere göre yerine 'Count' sorusunu ile otopark
butonlarının sayısını blok da yapabilirdim.

Öğrencinin imzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakem Yılı: 2007
Bilgi İşlem Ve Uygulama
Şefi

Yapılan İş: Araç giriş ve çıkış işlemleri

Tarih: 23/08/2018

Araç giriş ve çıkış için iki form oluştururdum.

Bu formlara sadece otopark durum içersindeki butonları tıkladığında giriş olucaktır. Başka herhangibir şekilde giriş mümkün almamaktır.

Araç giriş sayfasına giriş için butonun 'gold' yani içinin boş olması gerekmektedir. Araç giriş sayfası açıldığında otomatik olarak park yeri ve park saatı ekran'a gelecektir ve değiştirmeleri olucaktır. Ardından plaka ile müsteri kayıt kontrolü yapılacaktır. Eğer müsteri kayıtlı, değilse kayıt bilgilerini değiştirmek için müsteri eklene sayfasına yönlendirildim. Kayıtlı ise zaten bilgileri ekran'a değiştirmeleri olucaktır. Giriş yap butonuna basıldığında 'Insert into' komutu sayesinde veri tabanına kayıt oluşturulacak.

Araç giriş yapılan butonların rengi 'darkorange' olmak değişecek. Bu butonları tıkladığında ise araç çıkış formu ekran'a gelecektir. Araç çıkış formunda tam bilgiler veritabanından otomatik olarak ekran'a gelecektir. Bu sayfada eğer çıkış yap denilirse çıkış ekran'a gelecektir. Bu sayfada eğer çıkış yap denilirse çıkış yapılması için édeeme clma sayfasına gidecektir. Henüz bu formu oluşturmadım. Édeeme bosozlu alınırsa da 'delete' komutu sayesinde o cepte bulunan araç otoparktan silinecektir.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Üretim Uzmanı
Şefi

Araç çıkıştan gelen aracın otopark hizmetini kullandığı saatle göre elson hizmet ücreti, hizmet saatleri ekran da değiştirilemez olsak textboxta olmaktadır.

Bununla birlikte ödeme yapılmamışsa ve otomatik olarak oluşturulacaktır. Ödeme alınacak aracın plakası, park yeri ve araç

Schibinin adı da değiştirilemez olsak ekran da olmaktadır. Yani Öde alma formundan elle girilen veya değiştirilen bir

sey olmayacağı.

Form oluşturdukten sonra altına nihit ödeme ve kredi kartı
ödeme adında iki buton oluşturduk. Sındırılık sadece nihitle ödeme
alınabilmiş için kredi kartı ödeme al butonu tıklanmış ezellisini false
olarak değiştirdim.

Veri tabanında ödeme almak için oluşturduğum tabloda ödemeli
sayfası açıldığında onda bilgiler oluşuyor. Ancak ödeme iptal butonu
ile ödemeden vazgeçilirse veri tabanından 'delete' komutu ile veriler
silinerek güncellendiyor. İleride bu sayılabilir ve sorulabilecek
verileri kullanarak rapor oluşturmayı düşünüyorum.

Saatle göre ücret için tarife adında bir tablo oluşturup
icerisine beş farklı tarife ekledim. Dört tanesi saatlik olsak
ayrılandı, beşinci olan tarife ise günlük olarak yazıldı.

Tarife Bir = 0 - 1 saat = 7 TL

Tarife İki = 2 - 4 saat = 11 TL

Tarife Üç = 5 - 8 saat = 15 TL

Tarife Dört = 9 - 12 saat = 20 TL

Tarife Beş = Günlük = 25 TL

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakem Yazar
Bilgi İşleme
Şefi

Yapılan İş:

Tarih: 31/10/2018

Park İşlemleri	Ücret İşlemleri
Park Yeri:	Toplam Saat:
Giriş Tarihi:	Toplam Ücret:
Kişisel Bilgiler	
Ad Soyad:	
Telefon:	
Mail:	
Araç Bilgileri	
Plaka:	
Marka:	
Model:	
Renk:	
IS PARK İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.	

Plaka Kontrol	Park İşlemleri
Plaka:	Park Yeri:
	Giriş Tarihi:
Kişisel Bilgiler	
Ad Soyad:	
Telefon:	
Mail:	
Araç Bilgileri	
Plaka:	
Marka:	
Model:	
Renk:	
IS PARK İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.	

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin
Hakan YILMAZ
Bilgi İşlem ve Otoparkyon
Şefi

Yapılan İş: Araç Sayfası ile plaka kontrol

Tarih: 03/09/2018

Anasayfadan girilen bir araç soyadının formunu fesihledim. Formda plaka kontrolü yapmak için bir buton kaydum. Butonun içinde plaka sorgusu yapoken bu plakayı gösterilen tablodan değil park durum tablomdan yoptım. Bu sayede plakanın otopark hizmetini aktif olarak kullanıp kullanmadığını öğrendim. Eğer plaka eşleşmesi sağlanırsa ekran uyarı mesajı verdim. (Aracın şu an otoparkında değildir.)

Eğer araç plakası tablodaki plakalardan biriyle eşleşirse araçın otoparkın içerisinde olduğunu ve park yeri, park tarihi, rüteri ad, rüteri telefon, müsteri mail, araç plaka, araç markası, araç modeli, araç renk bilgilerini değiştirmeleri ekranı verdim.

```
string kontrol = "Select * from ParkDurum where parkmusteriadresid=@pl";  
SqlCommand cmd = new SqlCommand(kontrol, con.baglan +());  
SqlDataAdapter da = new SqlDataAdapter(cmd);  
SqlDataReader read = cmd.ExecuteReader();  
  
if (read.Read()) {  
    txtAdSoyad.Text = read["ParkMusteriAdSoyad"].ToString();  
    ...  
    ...  
    ...  
    txtAdSoyad.Enabled = false;  
    ...  
  
    MessageBox.Show("Girdiğiniz plaka otoparkındadır.");  
} else {  
    MessageBox.Show("Girdiğiniz plaka otoparkında değildir.");  
}
```

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

HAKAN YILMAZ
Bilgi İşlem ve Uygulamaları
Şefi

Yapılan İş: Admin Sayfası tasarımları ve giriş

Tarih: 04/10/2018

Önceden koordastirdığım gibi admin giriş'i için yeni bir form oluşturdum. Normal kullanıcı girişinden farklı olarak sayfa tasarımına uyacaktır. Admin sayfasından bu kullanıcıların kontrolü, rapor alma işlevleri ve tarife düzenlemeye işlevleri yelpilacaktır. Admin giriş'i için veritabanında yeni bir tablo oluştururdum. Bu tabloda admin ve parola verileri kayıtları edildi. Giriş sayfasında kodları güncelleyerek iki tabloda birden kontrol edilmesini sağladım. Admin ve parola eşleşmesi durumunda admin sayfası, kullanıcı adı ve parola eşleşmesi durumunda onasayfa, eşleşme olmaması durumda ise hata mesajı verilmistiir.

Admin onasayıta tasarımda ca yeri sekrete gitmek üzere içi içi pictureBox koydum. Benler kullanıcı işlevleri, tarife güncelleme, raporlar ve grafikler. Benlere uygun icerikler koydum ve tooltip ile tasarımi gizllestirdim.

Admin oturumu kapatması için de ufak bir pictureBox koydum ve tıkladığı zaman tekrar giriş sayfasına yönlendirdim.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakem YERBAZ
Bilgi İşlem ve İletişim
Şefi

Yapılan İş: Kullanıcı ekleme, silme, düzenlene

Tarih: 05/09/2018

Admin sayfasından kullanıcı ekleme, silme ve güncelleme işlevlerini
yapmak için öncelikle kullanıcılar formu ve bu formda dört pictureBox
yerleştirdim.

Kullanıcılar listesini veritabanından reader ile okuyarak datagrid'e
ekledim.

Insert into ile kullanıcı ekleme işlevlerini yaptım. Kullanıcı tablosunda
identity(1,1) olan bir kullanıcı id, kullanıcı adı ve şifre bulunuyordu.

Müşteri ekleme sayfasında yaptığım işlevleri tekrarladım.

Kullanıcı güncelleme ile istediğim zaman bir kullanıcıyı iptal
edebilir veya şifresini güncelleyebilirim.

```
private void btnSil_Click(object sender, EventArgs e) {  
    SqlCommand cmdsil = new SqlCommand("delete from Kullanicilar where  
    Kullaniciid=@dd", con.baglanti());  
    cmdsil.Parameters.AddWithValue("@dd", labelid.Text);  
    cmdsil.ExecuteNonQuery();  
    con.baglanti().Close();  
    MessageBox.Show("Kayıt Silindi", "Otopark");  
}
```

3

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve Teknoloji
Şefi

Yapılan İş:

Tarih: 05/09/2018

IS PARK
İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.

Kişisel Bilgiler

Kullanıcı Ad: _____
Parola: _____
T.C.: _____
Telefon No: (____) ____-____
Mail: _____
Adres: _____

Geri **Kaydet**

IS PARK
İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.

Kişisel Bilgiler

Kullanıcı Ad: _____
Parola: _____
T.C.: _____
Telefon No: (____) ____-____
Mail: _____
Adres: _____

Id

Güncelle **Sil**

Geri

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ
Bilgi İşlem ve İletişim
Şefi

Yapılan İş: Tarife Dözenleme

Tarih: 06/08/2018

Admin sayfasından yapılacak işleri bugün bitirmek için uğraştım. Kullanicı işlerini tamamen bitirmiştim. Eklene, dözenlene, silme ve isteneme işlerini halletmıştım.

Tarife Dözenleme formu oluşturarak veri tabanından 'update' komutyla güncelleme yaptıım. Bunun için eneslible tarife dözenleme sayfasındaki tarifeleri okuyarak ekranı getirdim. Burada tarifelere baktıktan sonra değiştirmek istediği tarifeyi seçerek değiştirebilim.

```
private void frmTarifeDuzenle_Load(object sender, EventArgs e){}
```

```
SqlCommand cmd=new SqlCommand("Select * From Tarifeler Where  
TarifeAd LIKE '%TarifeBir%'", con.baglanti());  
SqlDataReader oku =cmd.ExecuteReader();  
if(oku.Read())  
{  
    txtTarife1.Text=oku["TarifeBir"].ToString();  
    txtYesilTarife1.Text=oku["TarifeYesil"].ToString();  
}  
}
```

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakem Yazar
Bilgi İşlem ve
Şefi

Yapılan İş: Rapor, Grafik, Mali Durum

Tarih: 07/08/2018

Admin sayfasından aylık ve yıllık olarak rapor alınabilecek olacak.
Bu yazın admin sayfasından rapor ve grafikler pictureBox'ı ile oluşan
yeni formada 3 pictureBox yerleştirdim. MaliRapor, İstatistik ve
grafikler adında 3 yeni forma giderektir.

MaliRapor =

Kasayıllık rapor ve kasa aylık rapor alınabilecek tür. Yeni og ay
ne kadar kasaya para girdiğin bilgisi vermekinden almacaktır.

//Ay Toplam Kasa

```
String kontrol = "Select Sum(OdemeTopluUcret) from OdemeAll where  
OdemeAy=@p1" />Ort.Baglanti();
SqlCommand cmday = new SqlCommand(kontrol, con.baglanti());
```

Aynı zamanda senilen yıl ve oya göre kullanıcı tarife listeleride
ekranı verilecektir.

```
String tarifekontrol = "Select count(OdemeId) From OdemeAll where  
OdemeAy=@p1 and OdemeToplamSaat >= 0 and OdemeToplamSaat < 2";
SqlCommand cmd = new SqlCommand(tarifekontrol, con.baglanti());
cmd.Parameters.AddWithValue("@p1", cmbtarifeAy.Text);
SqlDataReader tarife = cmdtarife.ExecuteReader();
while (tarife.Read())
{
    tx + tarife[0].ToString();
}
con.baglanti().close();
```

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin
Hakem Yeri MAZ
Bilgi İşlem Uzmanı :
Şefi

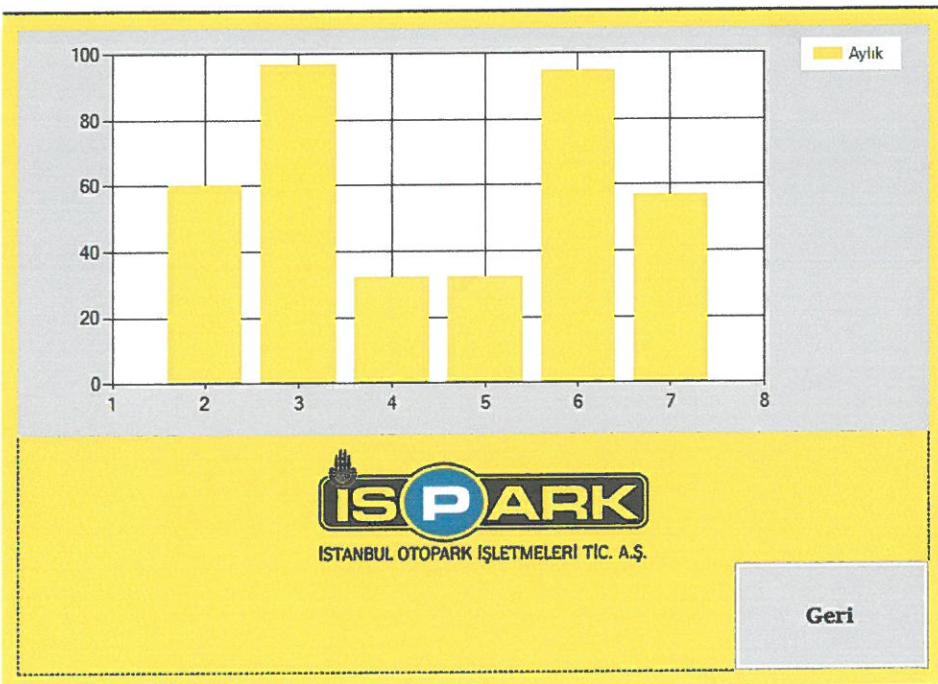
Yapılan İş:

Tarih: 07/03/2018

Kasa Yıllık Rapor Yıl Seçiniz: <input type="text"/> Seç Yıl: <input type="text"/>	Tarife Rapor Yıl Seçiniz: <input type="text"/> Ay Seçiniz: <input type="text"/>
Kasa Aylık Rapor Ay Seçiniz: <input type="text"/> Seç Ay: <input type="text"/>	Tarife Bilgileri 0-1 Saat: <input type="text"/> 2-4 Saat: <input type="text"/> 5-8 Saat: <input type="text"/> 9-12 Saat: <input type="text"/> Günlük: <input type="text"/>

ISPARK
İSTANBUL OTOPARK İŞLETMELERİ TİC. A.Ş.

Geri



Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

HAKAN M. MAZ
Bilgi İşlem ve Teknolojileri
Şefi

Yapılan İş: Açıls Animasyon, Simge, .exe yapımı

Tarih: 10/03/2018

Uygulama artık hazır hale geldikten sonra tasarım, iyileştirmek adına bir açılış animasyon sağfası yapmaya karar verdim.

Tüm formlarım 700; 500 ölçülerinde iken bu açılış animasyon sağfasının boyutunu 600; 350 olarak ayarladım.

Sadece logo kaydım bu sağfanın ekranında kalmıştı. Bu sonraki sayfada program.cs içinde bulunan kodları değiştirek uygulama ilk açıldığı anda bu sağfanın akması, kapanıkları sonra ise giriş sağfanının akmasını sağladım. Bu için timer kullandım.

private void timer1_Tick(object sender, EventArgs e)

{

Sayıett;

if(Sayıe<3){}

else {
timer1.Stop();
this.Close();
}

}

Bu islemleri de hallettikten sonra uygulamaya bir simge verme işlemi yaptıdım.

Otopark Otomasyonu → Özellikler → Uygulama → Simge ve Bildirim →

Gözat → Simge Seç → Kaydet.

Son işlem olarak da yaptığım bu uygulamanın bir .exe haliini makineye aldım.

Öğrencinin İmzası

Adı Soyadı :
İmzası :

Staj Amirinin

Hakan YILMAZ

Bilgi İşlem ve Teknoloji
Şefi

**SAKARYA ÜNİVERSİTESİ
MÜHENDİSLİK FAKÜLTESİ
STAJ GENEL ESASLARI**

I- GENEL İLKELER

Madde-1 : Stajlar, SAÜ Fakülteler, Yüksekokullar ve Devlet Konservatuarı Lisans Öğretim ve Sınav Yönetmeliği hükümleri gereğince öğrencilerin teorik bilgilerini pratikte pekiştirmek amacıyla yapmak zorunda oldukları uygulama çalışmalarıdır.

Madde-2 : Mühendislik Fakültesi öğrencileri 4 yıllık öğrenimleri süresince her bölümün özelliklerine göre düzenlenen staj programlarına uygun olarak, kesintisiz olarak 20'şer günlük (iş günü) iki kısımdan oluşmak üzere, **40 iş günü (8 hafta)** staj yapmak zorundadırlar.

Madde-3 : Staj işlemleri Fakülte düzeyinde Fakülte Staj Komisyonu'nda düzenlenir. Fakülte Staj Komisyonu, Fakülte Yönetim Kurulu'nda görevlendirilen 1 öğretim üyesi başkanlığında, Bölüm Staj Komisyonu Başkanlarından oluşur. Komisyon, Fakülte Staj Esasları'nın bölgümlerce uyum içinde yürütülmesini sağlar.

Madde-4 : Bölüm Staj Komisyonu, Bölüm Başkanının önerisi üzerine Fakülte Yönetim Kurulu'nda 2 yıl için görevlendirilen 1 başkan ve 2 veya 3 üyeden oluşur. Üyeler tekrar seçilebilirler.

Madde-5 : Stajlar öğretim ve sınav dönemlerini kapsayan süreler dışında ve Fakülte tarafından belirlenen tarihler arasında yapılır. Özel durumlarda öğrenim yılı içinde staj yapılabilmesi için öğrencinin **8 yarıyıllık** öğrenimini tamamlaması, Bölüm Staj Komisyonu'nun uygun görüşü ve Fakülte Yönetim Kurulu'nun kararı gereklidir.

Madde-6 : Stajlar bölgümlerin öngördüğü konularda ve belirlenen şartlarda resmi veya özel kurumlarda yapılır. Ancak, herhangi bir kurumda staj yapabilmek için stajın yapıldığı konuda uzman en az bir mühendisin o kurumda çalışıyor olması gereklidir.

Madde-7 : Aynı işletmede stajların tamamının yapılması Bölüm Staj Komisyonuna karara bağlanır.

Madde-8 : Fakülteye yatay geçiş veya dikey geçiş yolu ile gelen öğrencilerin geldikleri Fakülte veya Yüksekokulda yapmış oldukları stajın kabul edilmesi (en fazla 20 iş günü) Bölüm Staj Komisyonunun kararına bağlıdır.

Madde-9 : Öğrenciler, staj yapacakları işyerlerinin çalışma düzenleri ile ilgili kural, talimat ve emirlere uymakla yükümlüdür. Ayrıca Üniversitemize yakışır öğrencilik disiplini içerisinde olmalıdır, aksi halde stajları sayılmayabilir.

Madde-10 : Öğrenciler bölgümlerinin uygun gördüğü yerlerde staja başlayabilmeleri için ilgili kurumun onaylayacağı ve staj tarihleri belirtilen Staj Kabul Formlarını Dekanlık Staj Birimine teslim etmeden stajlarına başlayamazlar. Aksi durumda öğrencilerin sigorta işlemleri yapılamaz ve stajları kabul edilmez.

II- STAJ SÜRESİNCE YAPILACAK İŞLEMLER

Madde-11 : Staj defteri, Türkçe olarak, yazı ve resim kurallarına uygun şekilde mürekkepli kalemlle doldurulur. Defterde bulunan talimatlara uyulur ve bütün ilgili yerler doldurulur. Yurtdışında yapılan stajlara ait defterler İngilizce doldurulabilir. Öğrenci staj defterinde, Bölüm Staj Programı'nda istenen bütün hususları, programda belirtilen sıraya göre açıklamak zorundadır.

Madde-12 : Öğrenci, staj defterlerini işyerinde, staj süresi içerisinde doldurur. Çalışma süresinin sonunda, bütün sayfaların işyerindeki sorumlu mühendis tarafından imzalanması ve ilk sayfanın işyeri yetkilisi tarafından onaylı ve mühürlü olması gerekmektedir.

III- STAJ SONRASI YAPILACAK İŞLEMLER

Madde-13 : Staj defterleri Bölüm Staj Komisyonu'ncá belirtilen tarihlerde bizzat verilir ve bu sırada öğrenci yaptığı stajı komisyon karşısında sözlü olarak savunur.

IV- BÖLÜM STAJ KOMİSYONUNUN GÖREVLERİ

Madde-14 : Bölüm öğrencilerinin yapacağı tüm stajlar, Bölüm Staj Komisyonu'ncá planlanır, uygulamaya konulur ve denetlenir. Bölüm Staj Komisyonları her öğrencinin staj defterlerini inceleme ve sözlü sınav sonucunda yapılan stajın kabul ya da reddi konusunda karar verir.

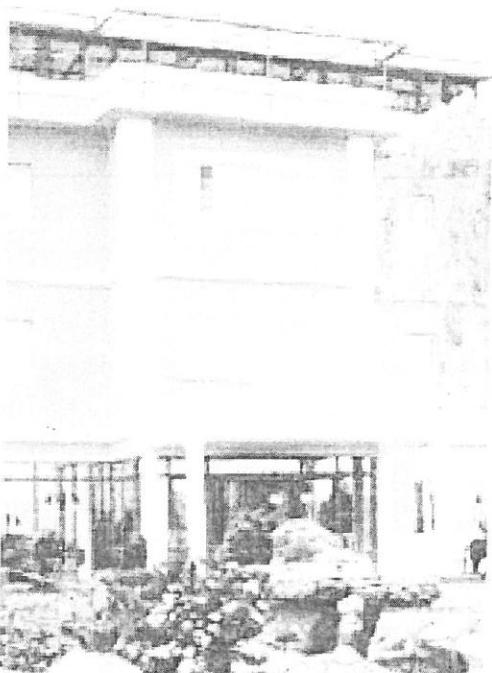
V- FAKÜLTE STAJ KOMİSYONUNUN GÖREVLERİ

Madde-15 :

- a) Fakülte öğrencilerine çeşitli kurum ve kuruluşlarca tahsis edilen staj yerlerini ilan etmek, yapılan başvuruları değerlendirmek ve bu yerlere öğrencilerin dağıtımını yapmak,
- b) Bölüm Staj Komisyonlarının yaptıkları staj değerlendirmelerine yapılacak itirazları inceleyerek Fakülte Yönetim Kurulu'na sunulmak üzere karara bağlamak,
- c) Fakülte genelinde stajlarla ilgili burada belirtilmeyen diğer problemlere çözüm bulmak.

Madde-16: İş bu esaslarda yapılacak değişikliler Fakülte Staj Komisyonunun teklifi ve Fakülte Kurulu'nun kararı ile yürürlüğe girer.

THE
CITY
OF
NEW
YORK



III
III

THE
CITY
OF
NEW
YORK



THE
CITY
OF
NEW
YORK